**Computer Science Key Stage 4 Curriculum Map (OCR J277 GCSE Computer Science)**

**Year 10**

| **Autumn 1** | **Autumn 2** | **Spring 1** | **Spring 2** | **Summer 1** | **Summer 2** |
| --- | --- | --- | --- | --- | --- |
| **Components covered:**  1.1 – Systems architecture  2.2 – Programming fundamentals (as part of Python programming) | **Components covered:**  1.2 – Memory and storage  2.2 – Programming fundamentals (as part of Python programming) | **Components covered:**  1.3 – Computer networks, connections and protocols  2.2 – Programming fundamentals (as part of Python programming) | **Components covered:**  1.4 – Network security  2.2 – Programming fundamentals (as part of Python programming)  2.4 – Boolean logic | **Components covered:**  2.2 – Programming fundamentals (as part of Python programming)  2.5 – Programming languages and Integrated Development Environments | **Components covered:**  2.2 – Programming fundamentals (as part of Python programming) |
| **Sub-Topics:**  1.1.1 Architecture of the CPU  1.1.2 CPU performance  1.1.3 Embedded systems  2.2.1 Programming fundamentals  2.2.2 Data types | **Sub-Topics:**  1.2.1 Primary storage (Memory)  1.2.2 Secondary storage  1.2.3 Units  1.2.4 Data storage  1.2.5 Compression  2.2.1 Programming fundamentals | **Sub-Topics:**  1.3.1 Networks and topologies  1.3.2 Wired and wireless networks, protocols and layers  2.2.3 Additional programming techniques | **Sub-Topics:**  1.4.1 Threats to computer systems and networks  1.4.2 Identifying and preventing vulnerabilities  2.2.3 Additional programming techniques  2.4.1 Boolean logic | **Sub-Topics:**  2.2.3 Additional programming techniques  2.5.1 Languages  2.5.2 The Integrated Development Environment (IDE) | **Sub-Topics:**  2.2.3 Additional programming techniques |
|  |  |  |  |  |  |
| **Assessment:**  End of unit tests.  PPE exams  Programming tasks | **Assessment:**  End of unit tests.  PPE exams  Programming tasks | **Assessment:**  End of unit tests.  PPE exams  Programming tasks | **Assessment:**  End of unit tests.  PPE exams  Programming tasks | **Assessment:**  End of unit tests.  PPE exams  Programming tasks | **Assessment:**  End of unit tests.  PPE exams  Programming tasks |
| **Builds upon:**  Hardware and Networks unit covered in Year 8 | **Builds upon:**  Data Representation unit covered in Year 9  Hardware and Networks unit covered in Year 8 | **Builds upon:**  Hardware and Networks unit covered in Year 8 | **Builds upon:**  Hardware and Networks unit covered in Year 8 | **Builds upon:**  Python programming covered in year 7, 8 and 9 | **Build upon:**  Python programming covered in year 7, 8 and 9 |
| **Introduces:**   * Von Neumann architecture * CPU registers | **Introduces:**   * Different types of secondary storage | **Introduces:**   * The Internet as a worldwide collection of computer networks:   + DNS (Domain Name Server)   + Hosting   + he Cloud   + Web servers and clients * Star and Mesh network topologies * Encryption * IP addressing and MAC addressing * Networking Standards * Common protocols including:   + TCP/IP (Transmission Control Protocol/Internet Protocol)   + HTTP (Hyper Text Transfer Protocol)   + HTTPS (Hyper Text Transfer Protocol Secure)   + FTP (File Transfer Protocol)   + POP (Post Office Protocol)   + IMAP (Internet Message Access Protocol)   + SMTP (Simple Mail Transfer Protocol) * The concept of layers | **Introduces:**   * Types of networks:   + LAN (Local Area Network)   + WAN (Wide Area Network) * Factors that affect the performance of networks * The different roles of computers in a client-server and a peer-to-peer network * The hardware needed to connect stand-alone computers into a Local Area Network:   + Wireless access points   + Routers   + Switches   + NIC (Network Interface Controller/Card)   + Transmission media * The Internet as a worldwide collection of computer networks:   + DNS (Domain Name Server)   + Hosting   + The Cloud   + Webservers and Clients * Star and Mesh network topologies | **Introduces:**   * The use of basic string manipulation * The use of basic file handling operations:   + Open   + Read   + Write   + Close * The use of records to store data * The use of SQL to search for data * The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional (2D) arrays * How to use subprograms (functions and procedures) to produce structured code * Random number generation * Characteristics and purpose of different levels of programming language:   + High-level languages   + Low-level languages * The purpose of translators * The characteristics of a compiler and an interpreter * Common tools and facilities available in an integrated development environment (IDE):   + Editors   + Error diagnostics   + Run-time environment   + Translators | **Introduces:**   * The use of basic string manipulation * The use of basic file handling operations:   + Open   + Read   + Write   + Close * The use of records to store data * The use of SQL to search for data * The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional (2D) arrays * How to use subprograms (functions and procedures) to produce structured code * Random number generation |

**Year 11**

| **Autumn 1** | **Autumn 2** | **Spring 1** | **Spring 2** | **Summer 1** | **Summer 2** |
| --- | --- | --- | --- | --- | --- |
| **Components covered:**  1.5 – Systems software | **Components covered:**  1.5 – Systems software | **Components covered:**  1.6 – Ethical, legal, cultural and environmental impacts of digital technology | **Components covered:**  2.1 – Algorithms | **Components covered:**  Revision:  Particular focus on exam technique and how to answer questions correctly. | **Components covered:** |
| **Sub-Topics:**  1.5.1 Operating systems | **Sub-Topics:**  1.5.2 Utility software | **Sub-Topics:**  1.6.1 Ethical, legal, cultural and environmental impact | **Sub-Topics:**  2.1.1 Computational thinking  2.1.2 Designing, creating and refining algorithms  2.1.3 Searching and sorting algorithms | **Sub-Topics:**  Revision:  Particular focus on exam technique and how to answer questions correctly. | **Sub-Topics:** |
| **Assessment:**  End of unit tests.  Practice exam questions | **Assessment:**  End of unit tests.  PPE exams | **Assessment:**  End of unit tests.  Practice exam questions | **Assessment:**  End of unit tests.  PPE exams | **Assessment:**  Practice exam questions on units 1 and 2 | **Assessment:** |
| **Builds upon:**  Parts of the Hardware and Networks unit covered in Year 8 | **Builds upon:**  Parts of the Hardware and Networks unit covered in Year 8 | **Builds upon:**  E-Safety lesson covered in KS3 | **Builds upon:**  Python programming covered in year 7, 8 and 9 | **Builds upon:**  Entire Computer Science course to date | **Build upon:** |
| **Introduces:**   * The purpose and functionality of operating systems:   + User interface   + Memory management and multitasking   + Peripheral management and drivers   + User management   + File management | **Introduces:**   * The purpose and functionality of utility software * Utility system software:   + Encryption software   + Defragmentation   + Data compression | **Introduces:**   * Impacts of digital technology on wider society including:   + Ethical issues   + Legal issues   + Cultural issues   + Environmental issues   + Privacy issues * Legislation relevant to Computer Science:   + The Data Protection Act 2018   + Computer Misuse Act 1990   + Copyright Designs and Patents Act 1988   + Software licences (i.e. open source and proprietary) | **Introduces:**   * Principles of computational thinking:   + Abstraction   + Decomposition   + Algorithmic thinking * Identify the inputs, processes, and outputs for a problem * Structure diagrams * Create, interpret, correct, complete, and refine algorithms using:   + Pseudocode   + Flowcharts   + Reference language/high-level programming language * Identify common errors * Trace tables * Standard searching algorithms:   + Binary search   + Linear search * Standard sorting algorithms:   + Bubble sort   + Merge sort   + Insertion sort | **Introduces:** | **Introduces:** |